

REMARKS

Claims 13-27 are pending and are unamended. Withdrawal of the outstanding rejection is respectfully requested for at least the reasons set forth below.

Entry of Rule 116 Response

Entry of this response is requested because this response does not raise any new issues that would require further consideration and/or search. No claim amendments are being presented in this response. No new matter is raised by this response. This response could not have been previously presented because the issues addressed herein are responsive to new grounds of rejection. Lastly, it is requested that the response be entered even if the application is not allowed because this response will place the application in better form for appeal by materially simplifying the issues.

If the application is not in proper form for allowance, Applicants request that the Examiner telephone the undersigned to discuss any further outstanding issues.

Request for Interview Prior to Formal Action on Amendment

Applicants request an interview prior to formal action on this response. An “Applicant Initiated Interview Request Form” accompanies this response. Please contact Applicants’ undersigned representative to schedule the interview.

Applicants have requested and conducted numerous interviews for this application and all of the previous interviews advanced prosecution of the application. The outstanding Office Action includes a new specification objection and a rejection based on a newly cited and applied reference (Loomans). Applicants believe that the nature of this case is such that the interview could serve to develop and clarify specific issues and lead to a mutual understanding between the Examiner and the Applicants, and thereby advance the prosecution of the application. Accordingly, it is believed that the purposes of an interview as outlined in MPEP 713.01, section III, will be fully served.

Examiner Interview

Applicants wish to thank Examiner Siddiqi for extending the courtesy of a personal interview in respect to this application on July 31, 2008 with Applicant and Applicants' undersigned representative. During the interview, a previously filed Amendment was discussed. No agreement was reached at the interview. Since the outstanding rejection presents only new grounds of rejections, no further comments regarding the interview are deemed to be necessary.

Specification Objection

The specification was objected for allegedly not providing a proper antecedent basis for the limitation "computer-readable medium" in claims 17-20. Applicants respectfully traverse this objection because the specification clearly supports this claim limitation and because the format of claims 17-20 is explicitly permitted by the USPTO as discussed in MPEP 2106.01, Section I.

Page 15, lines 22-26 of the present specification reads as follows:

The present invention can be included in an article of manufacture (e.g., one or more computer program products) having, for instance, computer useable media. The media has embodied therein, for instance, computer readable program code means for providing and facilitating the mechanisms of the present invention. The article of manufacture can be included as part of a computer system or sold separately.

This text portion clearly supports the claim limitation of a "computer readable medium." Furthermore, Appendices A-C and E show source code (i.e., computer-executable instructions) that become encoded on the computer-readable medium.

The Examiner states that the claimed "computer-readable medium encoded with computer-executable instructions" is interpreted as a computer program on a "shelf" that is waiting to be executed by a computer, and that this should be treated as a claim for a computer

program which is nonstatutory functional descriptive material. Applicants respectfully traverse this characterization of the claimed invention.

Claims 17-20 are directed to an "article of manufacture" which is a physical thing, such as a computer program product. For example, it may a diskette or a memory stick that contains software code, that when executed, performs the claimed steps. This is an entirely permissible claim format used to capture potential infringement by entities such as software retailers or distributors who are not performing the claimed process themselves, and are not selling hardware that meets an apparatus claim (e.g., software executing on a computer), but instead are selling software via an article of manufacture that performs the claimed process when loaded into a computer.

Furthermore, the Examiner's position is contrary to USPTO policy, as clearly stated in MPEP 2106.01, Section I (underlining added for emphasis).

Data structures not claimed as embodied in computer-readable media are descriptive material *per se* and are not statutory because they are not capable of causing functional change in the computer. See, e.g., *Warmerdam*, 33 F.3d at 1361, 31 USPQ2d at 1760 (claim to a data structure *per se* held nonstatutory). Such claimed data structures do not define any structural and functional interrelationships between the data structure and other claimed aspects of the invention which permit the data structure's functionality to be realized. In contrast, a claimed computer-readable medium encoded with a data structure defines structural and functional interrelationships between the data structure and the computer software and hardware components which permit the data structure's functionality to be realized, and is thus statutory....When a computer program is recited in conjunction with a physical structure, such as a computer memory, USPTO personnel should treat the claim as a product claim.

Applicants' claim language is explicitly permitted by the USPTO and cannot be characterized as nonstatutory functional descriptive material.

Rejection under 35 U.S.C. § 102

All pending claims were rejected under 35 U.S.C. § 102(e) as allegedly being anticipated by Loomans. Applicants respectfully traverse this rejection.

1. Loomans

Loomans discloses a process for deploying a generic application engine in a browser program executing on a client platform. Referring to Fig. 6, the process operates as follows:

1. A user loads the application engine shell page into a browser using an associated URL (step 602).
2. URL references within the frameset initiate loading of the kernel into the browser (step 604). The kernel is formed of code that is generally script embedded in HTML, such as Javascript or VB script. The Javascript (code) provides basic thread management functions such as event handling routines, blocking functions, and structures (such as frames) within which application components such as data and additional code may be loaded later, if required (column 5, lines 22-27).
3. The kernel, in turn, requests a minimum required subset of application engine components to load into the browser (step 606). These are the components that will be required for the user to proceed at the first page.
4. Once loaded, the application engine kernel runs any required initialization routines and initiates loading of any additional application engine components that are not required at starttime (step 608).
5. Concurrently with the bootstrapping¹ process in steps 604-608, the application engine shell page loads initial User Interface (UI) components and data components associated with the initial sub-application to be deployed (step 610).
6. Subsequently, the user provides requested inputs to the displayed UI components (step 612).
7. The application engine processes the user provided inputs in the particularized context of the sub-application's data and any API extension code (step 614).
8. The corresponding results are generated from the data and displayed in additional HTML in response to the user-supplied inputs (step 616).
9. A determination is made whether or not a new sub-application is to be processed (step 618).

¹ Bootstrapping refers to techniques that allow a simple system to activate a more complicated system.

10. If there are no additional sub-applications to process, the application simply waits for the next user input in the current sub-application. However, if there are additional sub applications to process, control is passed to step 620 where the additional components are loaded and control is then passed back to step 612.

2. Patentable differences between Loomans and claimed invention

Except for the fact that Loomans constructs a web page that includes script, Loomans has nothing else whatsoever to do with the claimed invention. (Applicants are not claiming to have invented constructing a web page that includes script.) The following table provides a snapshot view of the deficiencies in Loomans:

Claim language	Portions of Loomans highlighted by Examiner	Applicants' rebuttal comments
13. A method of constructing a web page that allows for syndication of digital assets, the method comprising:	Examiner asserts that Loomans provides this function.	No <u>digital assets</u> ² are syndicated in Loomans. In fact, digital assets are not discussed anywhere in Loomans.
(a) constructing a web page; and	Step 616 of Figure 6; column 9, lines 3-6	
(b) inserting into the web page script	embedded kernel 222 in Figure 2; column 6, lines 54-61	The Examiner is presumably referring to the fact that the kernel is formed of code that is generally <u>script embedded in HTML</u> , such as Javascript or VB script.
associated with at least one digital asset that is desired to be part of a fully rendered web page,	embedded kernel 222 in Figure 2; column 6, lines 54-61	As discussed above, the kernel and its associated script performs the following functions:

² A digital asset is any form of content and/or media that have been formatted into a binary source which include the right to use it. A digital file without the right to use it is not an asset. Digital assets are categorized in three major groups which may be defined as textual content (digital assets), images (media assets) and multimedia (media assets). Wikipedia definition of digital asset at: http://en.wikipedia.org/wiki/Digital_asset, attached hereto as an Appendix)

		<p>3. The kernel, in turn, requests a minimum required subset of application engine components to load into the browser (step 606). These are the components that will be required for the user to proceed at the first page.</p> <p>4. Once loaded, the application engine kernel runs any required initialization routines and initiates loading of any additional application engine components that are not required at starttime (step 608).</p> <p>The script in Loomans is <u>not</u> associated with at least one <u>digital asset</u> that is desired to be part of a fully rendered web page.</p> <p>Stated simply, the script in Loomans serves a completely different purpose than the script in the claimed invention.</p>
the inserted script including code to request the content of the digital asset from a remote site when the code is executed by a browser,	client computer 202 in Figure 2; column 6, line 36 through column 7, line 2	None of the functions performed by the kernel and its associated script request the <u>content of [a] digital asset</u> from a remote site.
the code including a uniform resource identifier (URI) of the web page	URL referred to on column 8, line 54	<p>The "URL references" on column 8, line 54 is not a URI of a <u>web page</u>. Instead, it a URL (location) of the kernel that is downloaded into the browser. A kernel is not a web page.</p> <p>There is no disclosure anywhere in Loomans of the concept of script code that includes a <u>URI (or its equivalent) of a web page</u>. This feature allows the remote site to verify that the request for the content of a digital asset has been made from a web page that has the appropriate permission to receive the content of the digital asset. In one preferred embodiment, the remote site compares the URI to an authorized list of URI's and only</p>

		<p>returns the content of the digital asset if the URI is authorized to receive it.</p> <p>Loomans has no authorization/permission step, so there is no reason to even add the concept of script code that includes a <u>URI (or its equivalent) of a web page</u> to Loomans.</p>
and a unique identifier of the content.	elements of Figure 6 (no specific elements were highlighted); column 8, lines 49-67	Loomans has nothing to do with "content," and thus the Figure 6 process described above inherently does not insert a unique identifier of content into the script code.

3. Patentability of independent claims 13, 17 and 21 over Loomans

Claim 13 reads as follows (underlining added for emphasis):

A method of constructing a web page that allows for syndication of digital assets, the method comprising:
 (a) constructing a web page; and
 (b) inserting into the web page script associated with at least one digital asset that is desired to be part of a fully rendered web page, the inserted script including code to request the content of the digital asset from a remote site when the code is executed by a browser, the code including a uniform resource identifier (URI) of the web page and a unique identifier of the content.

As discussed above in the table, Loomans does not disclose any of the above-highlighted features. Nor does Loomans have any disclosure that suggests such features. Accordingly, claim 13 is believed to be patentable over Loomans.

Claims 17 and 21 are also believed to be patentable over Loomans for the same reasons as applied to claim 13.

4. Patentability of dependent claims

The dependent claims are believed to be patentable over the applied references for at least the reason that they are dependent upon allowable base claims and because they recite additional patentable elements and steps.

5. Patentability of claims over prior art in outstanding EPO Office Action

Although not applied against the pending claims, such claims are believed to be patentable over the prior art references applied in the outstanding EPO Office Action submitted with the Supplemental Information Disclosure Statement filed on September 11, 2008. The following brief comments are provided regarding such references.

a. WO 00/20945 (Yee). Yee does not insert into a web page script associated with at least one digital asset. Nor does the use of COM objects in Yee have anything to do with this feature. Furthermore, since there is no script in Yee, there is no disclosure in Yee of the concept of script code that includes a URI (or its equivalent) of a web page. As discussed above, this feature allows the remote site to verify that the request for the content of a digital asset has been made from a web page that has the appropriate permission to receive the content of the digital asset.

b. U.S. Patent No. 5,999,941 (Andersen). Andersen also does not insert into a web page script associated with at least one digital asset. Andersen discloses the use of an applet (which is not script) within a browser of a personal computer 103. The applet prepares a URL to address an active server page (ASP) on a server computer 208. The “script” in Andersen is stored in the active server page (ASP) of the server computer 208. Also, the script in Andersen does not have code that includes a URI (or its equivalent) of a web page being constructed. Anderson has no authorization/permission step, so there is no reason to even add the concept of script code that includes a URI (or its equivalent) of a web page to Andersen.

In sum, neither of these references are relevant to the claimed invention.

Conclusion

Insofar as the Examiner's rejections were fully addressed, the instant application is in condition for allowance. Entry of this response, withdrawal of the Final Rejection, and issuance of a Notice of Allowability of all pending claims is therefore earnestly solicited.

Respectfully submitted,

RICHARD D. MARTIN et al.

November 26, 2008
(Date)

By: Clark Jablon
CLARK A. JABLON
Registration No. 35,039
PANITCH SCHWARZE BELISARIO & NADEL LLP
One Commerce Square
2005 Market Street - Suite 2200
Philadelphia, PA 19103
Telephone: (215) 965-1330
Direct Dial: (215) 965-1293
Facsimile: (215) 965-1331

Enclosure: Appendix (1 page)

APPENDIX

(attached to Response to Office Action of 8/18/08
for Application No. 09/923,923)

Digital asset

From Wikipedia, the free encyclopedia

Digital Asset: A digital asset is any form of content and/or media that have been formatted into a binary source which include the right to use it. A digital file without the right to use it is not an asset. Digital assets are categorised in three major groups which may be defined as textual content (digital assets), images (media assets) and multimedia (media assets); (van Niekerk, A.J. 2006).^[1]

In order to have a clear understanding of digital asset management the definitions of the different types of digital assets need to be defined and the difference must be specified. There are a number of management systems related to digital asset management (Austerberry, 2004) which are:

- a) Digital asset management (DAM).
- b) Digital content management (DCM).
- c) Enterprise content management (ECM).
- d) Digital media management (DMM).
- e) Media asset management (MAM).
- f) WEB content management (WCM).

An *art asset*, in computer graphics and related fields (particularly video game and visual effects production) is an individual piece of digital media used in the creation of a larger production. Art assets include synthetic and photographic bitmaps (often used for texture mapping, 3D models consisting of polygon meshes or curved surfaces), shaders, motion captured or hand-animated animation data, video and audio samples.

The term "art" is used to distinguish the creative (or real-world) elements of a production from the software or hardware used to create it: there is no requirement that the data represents anything artistic.

Digital asset management is expected to be a multi-billion dollar industry as corporations and individuals migrate traditional graphic, broadcast and print assets to the digital format. Companies including Apple, Oracle, Microsoft, Getty Images and others are aggressively expanding their enterprises to provide third-party digital asset management via web-based repositories. This trend will continue as business and consumers evolve from traditional analog materials.

See also

- digital asset management
- digital asset management system
- digital preservation

Retrieved from "http://en.wikipedia.org/wiki/Digital_asset"

Categories: Library and information science | Economics and finance stubs | Library and information science stubs

- This page was last modified on 6 October 2008, at 08:31.
- All text is available under the terms of the GNU Free Documentation License. (See **Copyrights** for details.)
Wikipedia® is a registered trademark of the Wikimedia Foundation, Inc., a U.S. registered 501(c)(3) tax-deductible nonprofit charity.